

日本特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日  
Date of Application:

2000年11月27日

出願番号  
Application Number:

特願2000-359553

出願人  
Applicant(s):

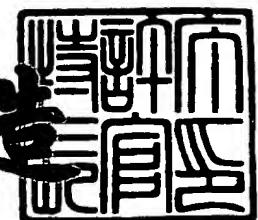
住友重機械工業株式会社

jc821 U.S. PTO  
09/986911  
11/13/01  


2001年 9月 4日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3080045

【書類名】 特許願  
【整理番号】 SJ0425  
【提出日】 平成12年11月27日  
【あて先】 特許庁長官 殿  
【国際特許分類】 B23K 26/06  
【発明者】  
【住所又は居所】 神奈川県平塚市夕陽ヶ丘63番30号 住友重機械工業  
株式会社 平塚事業所内  
【氏名】 西村 順也  
【特許出願人】  
【識別番号】 000002107  
【氏名又は名称】 住友重機械工業株式会社  
【代理人】  
【識別番号】 100080458  
【弁理士】  
【氏名又は名称】 高矢 諭  
【選任した代理人】  
【識別番号】 100089015  
【弁理士】  
【氏名又は名称】 牧野 勉博  
【選任した代理人】  
【識別番号】 100076129  
【弁理士】  
【氏名又は名称】 松山 圭佑  
【手数料の表示】  
【予納台帳番号】 006943  
【納付金額】 21,000円  
【提出物件の目録】  
【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9102448

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 加工計画方法及び装置

【特許請求の範囲】

【請求項1】

複数方向に走査又は移動可能とされた加工手段を用いてワークを加工する際に

ワーク上に散在する加工位置又は加工エリアを表わす点の位置データを、ツリー型のデータ構造で表現することを特徴とする加工計画方法。

【請求項2】

前記ツリーが、k次元二分探索木であることを特徴とする請求項1に記載の加工計画方法。

【請求項3】

ワーク上に散在する加工エリアの位置を決定した後、前記ツリーを用いて、各加工エリア内の点データを列挙することを特徴とする請求項1又は2に記載の加工計画方法。

【請求項4】

前記ツリーの根節点から、内部節点については、その節点の子節点が表わす領域と探索領域との重なり具合により、その節点の子節点を探索するかどうかを判定し、葉節点に行き着いた時のみ、直接、点データにアクセスすることを特徴とする請求項1に記載の加工計画方法。

【請求項5】

前記ツリーを用いて、注目点に最も近い点を探索する最近傍点探索処理を行うことを特徴とする請求項1又は2に記載の加工計画方法。

【請求項6】

前記最近傍点探索処理を、前記ツリーの根節点から始めることを特徴とする請求項5に記載の加工計画方法。

【請求項7】

注目点を中心とする、現在の最近傍点までの距離を半径を持つ円と節点の表わす領域が重なりを持つ場合に、その節点について探索を行うことを特徴とする請

求項6に記載の加工計画方法。

【請求項8】

前記節点が葉節点である場合は、当該節点が表わす領域内の全ての点について距離を求め、短いかどうか判定することを特徴とする請求項7に記載の加工計画方法。

【請求項9】

各点のデータが、それぞれ自分の所属する葉節点に関する情報を有し、前記最近傍点探索処理を、最近傍を探そうとしている注目点の存在する葉節点に直接アクセスし、次いで、探索が必要である節点が存在する場合に限り、前記ツリーの根節点の方向に溯ることにより行うことを行つことを特徴とする請求項5に記載の加工計画方法。

【請求項10】

注目点を中心とする、現在の最近傍点までの距離を半径に持つ円と節点の表わす領域が、現在の節点を表わす領域の外にはみ出す場合は、そのはみ出し具合に応じて、自分の兄弟節点又は自分の親の兄弟節点の探索を行うことを特徴とする請求項9に記載の加工計画方法。

【請求項11】

前記最近傍点探索処理と、見つかった最近傍点を外したツリーを用いて、次の最近傍点を探索する処理とを反復することにより、

注目点に近い順に近傍点を列挙することを特徴とする請求項5に記載の加工計画方法。

【請求項12】

前記最近傍点探索処理により、未だ加工経路につなげられていない最近傍点を探索して新たな注目点とし、加工経路につなげる処理と、

見つかった注目点を外したツリーを用いて、次の最近傍点を探索して新たな注目点とし、加工経路につなげる処理を、

始点から終点まで繰り返して、加工経路を作成することを特徴とする請求項5に記載の加工計画方法。

【請求項13】

前記ツリーから、見つかった点を外す処理を、各点データに付随するインデックスの終了番号を減らすか、又は、開始番号を増やすことにより行うことの特徴とする請求項11又は12に記載の加工計画方法。

【請求項14】

前記ツリーから、見つかった点を外すと、ある節点内の点データが全て消えてしまう場合は、その節点を訪れる必要が無いことを示す情報を、節点に持たせることの特徴とする請求項13に記載の加工計画方法。

【請求項15】

請求項12乃至14のいずれかに記載の加工計画方法により決定された加工経路を、巡回セールスマン問題を適用して加工順序を決定する際の初期解とすることの特徴とする加工計画方法。

【請求項16】

ある加工エリア内の点データの配置と、別の加工エリア内の点データの配置とが相対的に一致しているエリアの組を発見することにより、必要のない負荷の高い演算を省くことを特徴とする加工計画方法。

【請求項17】

前記点データの配置が相対的に一致するエリアの組を探す処理を、一方のエリアの全点に対し、一定の座標のずれ量を加えた点位置データを作成し、

もう一方のエリアのツリーの根節点から、前記点位置データの存在する領域を表す節点へと次々に移り、

葉節点に移ったときにのみ前記点位置データと一致する点位置データが存在するかどうかを調べるという処理により行うことの特徴とする請求項16に記載の加工計画方法。

【請求項18】

前記ツリーを、加工エリア内の点データを最少数の同寸矩形で囲む処理において、ループ呼び出しで一時的な加工エリアを作成する度に、その一時的な加工エリア、及び、各加工エリア内の一時的な点位置データの両方について作成するこ

とを特徴とする請求項1又は2に記載の加工計画方法。

【請求項19】

前記ツリーを、加工エリア確定時に、加工エリア、及び、各加工エリア内の点位置データの両方について作成することを特徴とする請求項1又は2に記載の加工計画方法。

【請求項20】

請求項1乃至19のいずれかに記載の加工計画方法により決定された加工を行うことを特徴とする加工方法。

【請求項21】

請求項1乃至19のいずれかに記載の加工計画方法を実施するためのコンピュータプログラム。

【請求項22】

請求項21に記載のコンピュータプログラムが記憶された、コンピュータ読取り可能な記録媒体。

【請求項23】

複数方向に走査又は移動可能とされた加工手段を用いてワークを加工する際に加工計画を立てるための加工計画装置において、

ツリー型のデータ構造で表現された、ワーク上に散在する加工位置又は加工エリアを表わす点の位置データを記憶する手段と、

前記ツリー型のデータ構造で表現された位置データを用いて、加工エリア位置や加工順序を決定する手段と、

を備えたことを特徴とする加工計画装置。

【請求項24】

請求項23に記載の加工計画装置を含むことを特徴とする加工装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数方向に走査又は移動可能とされた加工手段や、複数方向に移動可能とされたステージを用いて、該ステージ上に配置されたワークを加工する際

の加工計画方法及び装置に係り、特に、レーザビームを照射してプリント配線基板等に複数の穴開け加工を行うレーザ穴開け機に用いるのに好適な、穴開け等の加工位置の2次元平面における分布状態を数学的に捉えて、機器の動作を最適に計画することにより、加工時間を短縮する加工計画を行う際に、計画の精度を落とすことなく、計画にかかる計算時間を短縮することが可能な加工計画方法、該加工計画方法により決定された加工を行う加工方法、前記加工計画方法を実施するためのプログラムが記憶された、コンピュータ読み取り可能な記録媒体、同様の加工計画装置、及び、該加工計画装置を含む加工装置に関する。

#### 【0002】

##### 【従来の技術】

近年、電子機器の小型化や高密度実装化の要求に伴い、複数のプリント配線基板を重ね合わせた多層プリント配線基板が提供されるようになってきている。このような多層プリント配線基板では、上下に積層されたプリント配線基板のそれぞれに形成された導電層間を電気的に接続するため、これらの基板に、スルーホールあるいはビアホールと呼ばれる穴が形成される。そして、これらの穴の内部に導電膜を形成することにより、各プリント配線基板の導電層間の接続が行われる。

#### 【0003】

プリント配線基板に形成される穴は、最近のプリント配線基板の小型化や高機能化に伴って小型化し、直径0.1mm以下になってきている。このような小径の穴を精度よく形成するために、パルス発振型のレーザビームが用いられている。

#### 【0004】

従来のパルス発振型レーザを用いたレーザ穴開け機の一例の構成を図1（全体構成）及び図2（詳細構成）に示す。このレーザ穴開け機10は、パルス状レーザビームを発生するレーザ発振器12と、該レーザ発振器12により発生されたレーザビーム13の出力を、加工対象であるプリント配線基板（ワークとも称する）6の二つの走査エリア（加工エリア）8L、8Rに導いて、左右同時加工を可能とするために2等分するビームスプリッタ14と、該ビームスプリッタ14

により反射された左側のレーザビーム13Lを、走査エリア8L内でX軸（図の左右方向）方向及びY軸方向（図の前後方向）に走査するための左ガルバノユニット22L（図2参照）が収容された左ガルバノボックス20Lと、前記ビームスプリッタ12を通過した後、ミラー16で反射された右側のレーザビーム13Rを、右側の走査エリア8R内で同じくX軸方向及びY軸方向に走査するための右ガルバノユニット22R（図2参照）が収容された右ガルバノボックス20Rと、前記プリント配線基板8をX軸方向に平行移動するためのXステージ40X、及び、該Xステージ40X上で前記プリント配線基板8をY軸方向に移動するためのYステージ40Yを含むXYステージ40（図2参照）とを主に備えている。

#### 【0005】

前記左ガルバノボックス20L及び右ガルバノボックス20Rには、図2に詳細に示す如く、レーザビーム13L、13Rをそれぞれ反射するためのミラー24L、24Rと、該ミラー24L、24Rで反射されたレーザビームを、例えばY軸方向に走査するための第1ガルバノミラー26L、26Rと、該第1ガルバノミラー26L、26Rを駆動するためのガルバノドライバが内蔵された第1ガルバノスキャナ（単に第1スキャナとも称する）28L、28Rと、前記第1ガルバノミラー26L、26RによってY軸方向に走査されたレーザビームを、更にこれに垂直なX軸方向に走査するための第2ガルバノミラー30L、30Rと、該第2ガルバノミラー30L、30Rを駆動するためのガルバノドライバが内蔵された第2ガルバノスキャナ（単に第2スキャナとも称する）32L、32Rと、前記第1及び第2ガルバノミラー26L、26R、30L、30RによりX軸方向及びY軸方向に走査されたレーザビームを、プリント配線基板8の表面に對して垂直に偏向し、照射口（図示省略）を介して落とすためのfθレンズ34L、34Rとを備えた左ガルバノユニット（単に左ユニットとも称する）22L、及び、右ガルバノユニット（単に右ユニットとも称する）22Rが、それぞれ収容されている。

#### 【0006】

ここで、例えば前記左ガルバノボックス20Lは固定され、右ガルバノボック

ス20Rは、例えば、そのX軸方向位置が可変とされ、加工開始前に、ユニット間距離（L軸値と称する）Aが変更可能とされている。

#### 【0007】

基板6上のビーム照射可能範囲は、fθレンズ34L、34Rの大きさがコストや品質等の点で制限されているため、加工中は完全に位置が固定されるビーム照射口の真下位置を中心とする、例えば40mm×4mmのX軸、Y軸に両辺が平行な矩形範囲（走査エリアと称する）8L、8Rに限定されている。

#### 【0008】

一方、基板6の大きさは、例えば最大約50mm×600mm程度の大きさであり、一般に走査エリアよりも広い。従って、基板全体の穴開け加工を行うために、加工中はガルバノユニットの位置を固定したまま、基板を支えるXYステージ40をXY平面内で自由な方向に駆動させ、基板6を移動するようにしている

#### 【0009】

このようなレーザ穴開け機は、図3に全体の動作を示す如く、

- (1) XYステージ40によって基板6を移動する、
- (2) 左右の走査エリア8L、8R内のレーザ穴開け加工を行う、

という2ステップの繰り返し（ステップアンドリピートと称する）により、基板全体の穴開けを行っている。被加工面では、レーザビームが照射された部分が蒸発し、プリント配線基板6に穴が形成される。ここで、レーザビームは2分されているため、同時に加工される走査エリアは左右に2つあり、同時加工エリア間距離は加工ヘッド間距離に等しい。

#### 【0010】

又、各走査エリア8L、8R内の加工については、図4に示す如く、左右ユニット22L、22Rの第1、第2スキヤナ28L、28R、32L、32Rによる、基板上の穴開け終了点から次の穴開け予定点への移動走査（ビーム走査と称する）が全て完了した段階でレーザビームを照射するという工程を繰り返している。

#### 【0011】

以下、レーザ穴開け機のシステム構成を極めて簡単に示した図5に従って、システム構成について説明する。

#### 【0012】

必要な入力データ50は、穴開け位置データを主とするCADデータ、及び、例えばパソコン上のウインドウから入力されるGUI入力である。加工計画装置52は、加工計画に必要なデータを用いて加工計画を実行し、加工計画により作成もしくは順番を変更されたデータ、及び、加工計画には関係しないデータから、所定の形式の複数個のファイルによって構成される変換データファイル群54を作成する。この変換データファイル群54により、レーザ穴開け機の穴開け加工実処理部60が動作する。

#### 【0013】

レーザ穴開け機は、簡単に言うと、制御装置62と、各内部機器64からなる。各内部機器64は、加工開始直前に所定の位置までX軸に平行にシフト移動する前記ガルバノ右ユニット22Rと、前記レーザ発振器12と、合計4つの前記ガルバノスキャナ28L、28R、32L、32Rと、前記XYステージ40とから構成されており、前記制御装置62が各内部機器64に駆動信号を送ると、各内部機器64が駆動され、各内部機器64の移動が完了すると、完了信号を制御装置62に返すようにされている。

#### 【0014】

本発明は、出願人が特願2000-345417で提案した加工計画装置52の計画自体を高速化しようとするためのものであり、本発明が有る無しに拘らず、アルゴリズムの大きな流れに変化はない。そこで、前記加工計画装置52の構成及び処理手順を示す図6に従って、加工計画装置自体の処理内容、大まかなフロー及び各アルゴリズムについて簡単に説明する。なお、処理の内容の詳細は特願2000-345417に記載したとおりである。

#### 【0015】

まず、処理100の内部ループ処理102では、図7に示す如く、「分割線に分けられた領域の一方を、例えば（分割線）と、（分割線と平行な矩形枠の辺2つのうちの一方）とが一致するように平行シフトさせて2つの領域を重ね、（点

配置) と (点の存在する矩形枠) という情報をもつた一時的な領域 (テンポラリ領域と称する) を作る」処理と、「矩形枠内の全ての点を、最少数の互いに平行な同寸矩形により囲む」処理を行う。ここで、点は穴開け位置に相当し、矩形枠がビーム照射領域、同寸矩形が走査エリアに相当する。この処理により、L軸値に対する、ステージ移動回数及びビーム走査回数という、加工時間に基づく2つのデータを求めることができる。

#### 【0016】

前記内部ループ処理102を含む処理100は、図8に示す如く、「一変数関数の最小化問題」の解法処理である。「一変数関数の最小化問題」とは、変数X、Yに対して、関数Fを用いて  $Y = F(X)$  で表わせる時の、Yが最小値を採る時のXの値を求める問題である。ここでは、X軸にL軸値(ユニット間距離)を探り、関数Fとして、処理102における「矩形枠内の全ての点を、最少数の互いに平行な同寸矩形により囲む方法」の解法処理をそのまま用い、関数Fが返す値、即ちY軸には、最適な状態を表わすのに適切な評価値、即ち「ステージ移動回数S」及び「ビーム走査回数G」に、実験により得られた相応しい重みパラメータ $\beta$ 、 $\gamma$ をかけたもの、 $(\gamma \times G + \beta \times S)$ を採る。

#### 【0017】

前記処理102を内部に含む処理100により、ユニット間距離(L軸値)と、走査エリアの大まかな位置のデータが作成される。

#### 【0018】

前記処理100に続く処理104は、図9及び図10に示す如く、走査エリアの左右の穴数がなるべく均等になるように配分する処理と、図11に示す如く、走査エリアの位置を、穴開け位置がなるべくエリアの中心に来るよう微調整する処理の2つである。

#### 【0019】

前記穴数を配分する処理では、図9に示す如く、エリアFとGが重複していて、エリアG(1)、G(2) (括弧内の数字1は左ユニット、数字2は右ユニットを表わす)の両方に重複する領域W(1)、W(2)がある場合に、次式に示す如く、左右のエリアFの穴数の多い方と、左右のエリアGの穴数の多い方の和

が最小となるよう、左右の同時加工エリアの穴数を均等に配分することによって、時間短縮につなげることができる。

【0020】

$$\text{Min}(\text{Max}(F(1) \text{の穴数}, F(2) \text{の穴数}) + \text{Max}(G(1) \text{の穴数}, G(2) \text{の穴数})) \quad \dots (1)$$

【0021】

今、図10の上段に示す如く、左ユニットのエリアF(1)のみに属する穴開け位置が30穴、同じくエリアG(1)のみに属する穴開け位置が20穴、重複エリアW(1)が60穴とし、右ユニットのエリアF(2)のみに属するものが10穴、エリアG(2)のみが40穴、重複エリアW(2)が50穴であったとすると、特に工夫することなく、機械的に、例えば、先に決定したエリアに全て配分した場合には、図10の中段に示す如く、左ユニットのF(1)が90穴、G(1)が20穴、右ユニットのF(2)が60穴、G(2)が40穴となり、エリアFのガルバノショット数が90回、エリアGのガルバノショット数が40回で合計130回となる。

【0022】

これに対して、同時加工エリアの穴数を均等化した場合には、図10の下段に示す如く、左ユニットのF(1)に55穴、G(1)に55穴、右ユニットのF(2)に55穴、G(2)に45穴で、エリアFのガルバノショット数が55回、エリアGのガルバノショット数が55回となり、合計110回で、計20回減少させることができる。

【0023】

又、走査エリアの位置を、穴開け位置がなるべくエリアの中心に来るよう微調整する処理では、図11に示す如く、例えば走査エリアの中心を、元のエリア8内の点データの存在位置を使って微調整し、穴開け位置が、なるべく修正後の走査エリア8の中央に乗るようにして、加工精度を向上する。

【0024】

この操作は時間短縮には結び付かないが、fθレンズ34L、34Rの収差等の関係で、中心の方が高精度であるため、加工精度向上に有効である。

## 【0025】

前記処理104により、走査エリアの正確な位置のデータが作成され、又、各穴開け位置が所属する走査エリアが、各穴開け位置に対して1つ定められる。

## 【0026】

処理104が終了した後、処理106に進み、走査エリアの訪問順序（ステージパスと称する）を最適化して確定する。具体的には、一般に広く知られている巡回セールスマン問題（全ての都市を訪問して最初の都市に戻ってくる巡回経路の中で、一巡経路長が最小のものを求める問題：TSPとも略する）、もしくは、状況に応じて一般的なTSPを改良した方法（改良TSPと称する）を適用する。

## 【0027】

状況に応じた巡回セールスマン問題の適用法及び始点と終点の決定に際しては、例えば、図12に示す如く、巡回セールスマン問題を適用して一巡訪問順序を決定し、最も距離の長い（時間のかかる）移動Lmaxを除くよう、始点と終点を決定することができる。

## 【0028】

あるいは、実用上、特に訪問位置が少ない時等は、代替処理として、図13に示す如く、「（一巡経路長）-（最長の移動長）」を最短にするように一巡経路を決定して、最長の移動を除くようにすることもできる。巡回セールスマン問題の手法のうち、kオプト法やLK法は、いずれも「解（ある巡回経路）の近傍を探索し、近傍解で更新する」というループの繰り返しにより解を逐次改善していく。従って、各ループの「解」として、「一巡経路長」ではなく、「（一巡経路長）-（最長の移動長）」で評価すれば、最終的な解としても、前記「（一巡経路長）-（最長の移動長）」が最小になる。図13の場合、右側の方が一巡経路長は少ないが、「（一巡経路長）-（最長の移動長）」で比べると、左側の方が最適解となっている。

## 【0029】

この処理106により、走査エリアの訪問順序（ステージパスとも称する）が確定する。処理106で確定した各走査エリアに対して、穴開け位置の訪問順序

(ガルバノパスとも称する)を処理108で計算する。この処理108においては、処理106における処理と同様のセールスマン問題又は改良セールスマン問題による処理に加えて、更に、マッチングを探る。具体的には、図14の上段に示す如く、左右のユニットの同時加工エリアのビーム走査時間に偏りがあって、一方に待ち時間が生じる場合には、始点を互いにずらしてマッチングを取ることによって、ビーム走査時間の偏りをなくし、総加工時間を短縮することができる。

#### 【0030】

この処理108により、各走査エリア内の穴開け位置訪問順序のデータが作成される。

#### 【0031】

##### 【発明が解決しようとする課題】

前記のような計画装置の大きな特徴として、巡回セールスマン問題等、組合せ最適化問題的な面と、二次元平面のデータの探索等、計算幾何学的な面がある。これらの問題は一般に、処理方法が悪いと、膨大な計算時間を必要とする。即ち、穴開け位置というような幾何学的点データを解析し計画する場合、一般的に、計画の精度を向上することと計画にかかる計算(CPUによる演算)時間とはトレードオフの関係にあり、良い計画を実現するためには、計画の実行形態がオンライン処理であれオフライン処理であれ、計算に費やす時間が膨大に必要となり、計画自体に非常に時間が必要になるという問題点を有していた。

#### 【0032】

本発明は、前記従来の問題点を解消するべくなされたもので、幾何学的点データを効率良く格納するデータ構造を導入することにより、加工時間短縮を目的とする計画の精度を落とすことなく、計画にかかる計算時間を短縮することを第1の課題とする。

#### 【0033】

本発明は、又、前記加工計画により決定された加工を行う加工方法を提供することを第2の課題とする。

#### 【0034】

本発明は、又、前記加工計画を実施するためのコンピュータプログラムを提供することを第3の課題とする。

【0035】

本発明は、更に、前記コンピュータプログラムが記憶されたコンピュータ読取り可能な記録媒体を提供することを第4の課題とする。

【0036】

本発明は、又、前記加工計画を実施するための加工計画装置を提供することを第5の課題とする。

【0037】

本発明は、又、前記加工計画装置を含む加工装置を提供することを第6の課題とする。

【0038】

【課題を解決するための手段】

本発明は、複数方向に走査又は移動可能とされた加工手段を用いてワークを加工する際に、ワーク上に散在する加工位置又は加工エリアを表わす点の位置データを、ツリー型のデータ構造で表現することにより、前記第1の課題を解決したものである。

【0039】

又、前記ツリーを、k次元二分探索木としたものである。

【0040】

又、ワーク上に散在する加工エリアの位置を決定した後、前記ツリーを用いて、各加工エリア内の点データを列挙するようにしたものである。

【0041】

又、前記ツリーの根節点から、内部節点については、その節点の子節点が表わす領域と探索領域との重なり具合により、その節点の子節点を探索するかどうかを判定し、葉節点に行き着いた時のみ、直接、点データにアクセスするようにしたものである。

【0042】

又、前記ツリーを用いて、注目点に最も近い点を探索する最近傍点探索処理を

行うようにしたものである。

【0043】

又、前記最近傍点探索処理を、前記ツリーの根節点から始めるようにしたものである。

【0044】

又、注目点を中心とする、現在の最近傍点までの距離を半径に持つ円と節点の表わす領域が重なりを持つ場合に、その節点について探索を行うようにしたものである。

【0045】

又、前記節点が葉節点である場合は、当該節点が表わす領域内の全ての点について距離を求め、短いかどうか判定するようにしたものである。

【0046】

又、各点のデータが、それぞれ自分の所属する葉節点に関する情報を有し、前記最近傍点探索処理を、最近傍を探そうとしている注目点の存在する葉節点に直接アクセスし、次いで、探索が必要である節点が存在する場合に限り、前記ツリーの根節点の方向に溯ることにより行うようにしたものである。

【0047】

又、注目点を中心とする、現在の最近傍点までの距離を半径に持つ円と節点の表わす領域が、現在の節点を表わす領域の外にはみ出す場合は、そのはみ出し具合に応じて、自分の兄弟節点又は自分の親の兄弟節点の探索を行うようにしたものである。

【0048】

又、前記最近傍点探索処理と、見つかった最近傍点を外したツリーを用いて、次の最近傍点を探索する処理とを反復することにより、注目点に近い順に近傍点を列挙するようにしたものである。

【0049】

又、前記最近傍点探索処理により、未だ加工経路につなげられていない最近傍点を探索して新たな注目点とし、加工経路につなげる処理と、見つかった注目点を外したツリーを用いて、次の最近傍点を探索して新たな注目点とし、加工経路

につなげる処理を、始点から終点まで繰り返して、加工経路を作成するようにしたものである。

#### 【0050】

又、前記ツリーから、見つかった点を外す処理を、各点データに付随するインデックスの終了番号を減らすか、又は、開始番号を増やすことにより行うようにしたものである。

#### 【0051】

又、前記ツリーから、見つかった点を外すと、ある節点内の点データが全て消えてしまう場合は、その節点を訪れる必要が無いことを示す情報を、節点に持たせるようにしたものである。

#### 【0052】

又、前記のいずれかの加工計画方法により決定された加工経路を、巡回セールスマント問題を適用して加工順序を決定する際の初期解とするようにしたものである。

#### 【0053】

又、ある加工エリア内の点データの配置と、別の加工エリア内の点データの配置とが相対的に一致しているエリアの組を発見することにより、必要なない負荷の高い演算を省くことにより、前記第1の課題を解決したものである。

#### 【0054】

更に、前記点データの配置が相対的に一致するエリアの組を探す処理を、一方のエリアの全点に対し、一定の座標のずれ量を加えた点位置データを作成し、もう一方のエリアのツリーの根節点から、前記点位置データの存在する領域を表す節点へと次々に移り、葉節点に移ったときにのみ前記点位置データと一致する点位置データが存在するかどうかを調べるという処理により行うようにしたものである。

#### 【0055】

又、前記ツリーを、加工エリア内の点データを最少数の同寸矩形で囲む処理において、ループ呼び出しで一時的な加工エリアを作成する度に、その一時的な加工エリア、及び、各加工エリア内の一時的な点位置データの両方について作成す

るようとしたものである。

【0056】

又、前記ツリーを、加工エリア確定時に、加工エリア、及び、各加工エリア内の点位置データの両方について作成するようにしたものである。

【0057】

本発明は又、前記のいずれかの加工計画方法により決定された加工（例えばレーザ加工やドリル加工）を行うようにして、前記第2の課題を解決したものである。

【0058】

本発明は、又、前記のいずれかの加工計画を実施するためのコンピュータプログラムにより、前記第3の課題を解決したものである。

【0059】

本発明は、又、前記コンピュータプログラムを記憶することにより、前記第4の課題を解決したものである。

【0060】

本発明は、又、複数方向に走査又は移動可能とされた加工手段を用いてワークを加工する際に加工計画を立てるための加工計画装置において、ツリー型のデータ構造で表現された、ワーク上に散在する加工位置又は加工エリアを表わす点の位置データを記憶する手段と、前記ツリー型のデータ構造で表現された位置データを用いて、加工エリア位置や加工順序を決定する手段とを備えることにより、前記第5の課題を解決したものである。

【0061】

本発明は、又、加工装置が、前記加工計画装置を含むことにより、前記第6の課題を解決したものである。

【0062】

【発明の実施の形態】

以下図面を参照して、本発明の実施形態を詳細に説明する。

【0063】

本発明は、2次元平面における点の位置データを、ツリー型のデータ構造、例

えば  $k$  次元二分探索木（略して  $k$  d - 木）で表現することにより、例えばレーザ穴開け機の加工計画装置に表われる、次のような処理を高速化し、計画自体にかかる時間を短縮する。

## 【0064】

(1) 矩形領域内に存在する点を列挙する矩形領域探索問題

## 【0065】

(2) ある点について最も近い点を探す最近傍点探索問題

## 【0066】

(3) ある点についての近傍点を、ある個数まで順番に挙げていく近傍点リストの作成

## 【0067】

(4) 巡回セールスマン問題の解法の1つであるニアレストネイバー法による巡回経路の形成

## 【0068】

(5) ある領域とある領域の点配置が、全く同じかどうかを判定する全一致問い合わせ処理

## 【0069】

まず、 $k$  d - 木について説明する。 $k$  d - 木は、多次元空間内の点データの探索（例えば、データベースでの基本的な問い合わせである、直交領域探索（range query）、全一致問い合わせ（exact match query）、部分一致問い合わせ（partial match query）など）を効率よく行うためのデータ構造の一つであり、一次元領域内の探索を効率よく行うためのデータ構造として知られている、二分探索木（binary search tree）を多次元に一般化したもので、グラフ理論における二分木の構造に表現されるものの一つである。

## 【0070】

ここで、図15に示す二分木に関して、用語についてのみ簡単な注訳を加えておく（説明のための最低限の注訳であり、用語の正確な定義についてはグラフ理論の参考書に委ねたい）。二分木は通常、最上部に配置された根（root（図中のR））と呼ばれる特別な節点（node（○印））から系統的に下方へ広がった形

に表される。それゆえ、二分木の節点同士の関係は、一般的に生物の血縁関係を表す用語を用いて表される。例えば、AはCの「親」、D、EはBの「子」、DとEとは「兄弟」、AはFの「祖先」、FはAの「子孫」など。二分木の定義から、各節点は最大2つの子節点しか持たない。また、子を持たない節点を「葉（leaf）」節点といい、その他の節点を内部（internal）節点（根を含む）と呼び区別する。

## 【0071】

通常の二分探索木は、簡単に言うと、

- (1) 木の内部節点（根を含む）は、必ず2つの子節点を持ち、探索を手助けするための分割値を蓄える、
- (2) 木の葉節点は、点そのもの（複数も可）を蓄える、

という構造を有する。

## 【0072】

この考えを拡張し、多次元の二分探索木、 $k$  d - 木の場合には、

- (1) 木の内部節点（根を含む）は、2つの子節点を持ち、探索を手助けするための分割線を入れる軸（X軸又はY軸）及び分割値を蓄える、
- (2) 木の葉節点は、点そのもの（複数も可）を蓄える、

という構造を有する。即ち、通常の二分探索木では、内部節点は、区間情報（線分）と二分するための分割値（線分の切断箇所）を保持していたが、 $k$  d - 木の場合は、領域情報（矩形領域）と領域を二分するための分割線（矩形の縦もしくは横に平行な二分直線）を保持する。例えば、図16のように二次元平面に散布する9個の点データ（P1～P9）は、二分木を用いると図17のように表現され得る。各点データは、根から順に、内部節点の蓄える分割線により次々に二分され、最終的に点データの個数が二個以下にまで細分されると、葉節点となっているのが分かる。

## 【0073】

2次元  $k$  d - 木は、分割軸の設定の仕方等にバリエーションがあるが、本実施形態では、次のように定めている。

## 【0074】

(1) 内部節点(図17の11、12、13、14)における分割軸は、その節点の表わす矩形領域(節点の表す矩形領域とは、元の点データの存在する領域が根から順に次々と二分割されるときの、その節点の蓄える分割線によって二分される前の領域のことを表す。例えば、根の表す領域とは、元の点データの広がりそのものである)の、横幅と縦幅の広がりを比較し、大きい方を二分するように設定する。

## 【0075】

(2) 内部節点における分割値は、その節点の表わす矩形領域の全ての点( $n$ 個とする)を、分割軸の値(分割軸がY軸なら、Y座標のことを指す)が、小さい方から数えて、 $n/2$ 番目( $n$ が奇数である場合は切り捨て)の要素(即ち、真中の要素)の値により分割する。

## 【0076】

(3) 節点は、その節点の示す点データの数がある個数以下(図17では2個以下)になったとき葉節点(図17ではP1・・・P9)となる。

## 【0077】

(4) 従って、葉節点は、

図18に示す如く、(インデックス付の配列の点データが格納されているとして)点データに割り振られた番号のインデックスを2つ持つ(始まり位置と終わり位置)。

## 【0078】

各節点は、具体的には、例えばC言語(C++言語)における構造体を用いると、図18のように表すことができる。ここで、`struct kdnode`は節点、`bucket`(整数型)は葉節点(例えば1)と内部節点(例えば0)の別、`cutdim`(整数型)は分割軸の方向(X軸又はY軸に平行)、`cutval`(浮動小数点型もしくは整数型)は分割値、`*lson`、`*hson`は子節点のポインタ、`lopt`(整数型)は開始インデックス、`hipt`(整数型)は終了インデックスを、それぞれ示す。また、各葉節点の表す領域に所属する実際の点データは、例えば配列を用いて`perm[lopt..hipt]`に蓄えられる。

## 【0079】

kd-木において、各節点が保持すべき最低限必要な項目は、次のとおりである。

【0080】

- ・内部節点・・・分割線（分割軸及び分割値）及び子の情報

- ・葉節点・・・穴開け位置の開始インデックスと終了インデックス

従って、内部節点（bucket=0）の場合は、図18のcutdim、cutval、\*loson、\*hisonがあれば良く、葉節点（bucket=1）の場合は、loptとhiptがあれば良い。

【0081】

以下、レーザ穴開け機の加工計画装置において、kd-木を用いることによる効果のある処理について、説明する。

【0082】

（1）矩形領域探索

図19に示す如く、矩形領域範囲（枠）内の、枠に平行な矩形領域内の点を列挙する矩形領域探索処理における、「領域内の点を最少数の同寸矩形で囲む」処理において、エリア位置決定過程で、又、エリア位置を決定後、そのエリア内の点を列挙するときに用いる。

【0083】

即ち、従来は、全ての点データにアクセスし、探索領域内か否かを判定し、報告していたのに対し、本発明では、ツリーの根節点から、葉節点へのトップダウン的な探索を行う。即ち、節点が内部節点であるならば、その節点の表わす領域と探索領域との重なり具合により、その節点の両方の子節点を探索するかどうかを判定する。判定は、次のようにして行う。

【0084】

（1）探索領域が節点の表わす領域内に含まれるとき・・・その節点を根とする部分木内の点データ全てを報告する。

【0085】

（2）探索領域と節点の表わす領域とが一部重なるとき・・・探索を続ける。

【0086】

(3) 探索領域と節点の表わす領域とが重なりを持たないとき・・・探索しない。

【0087】

節点が葉節点であれば、その節点の全ての点データにアクセスし、探索領域内かどうかを判定して、探索領域内の点を順に報告する。

【0088】

従来の方法では、全ての点にアクセスする必要があり、レーザ穴開け機の穴開け位置の数が多くなるにつれ、探索にかかる時間が大きく膨れ上がるのに対して、本発明に係る  $k$  d-木による探索方法では、直接点データにアクセスするのではなく、木の根節点から葉節点へのトップダウン的な探索方法であり、各節点での判定は、探索領域との重なり具合（包含、重複、重複無し）のみであり、葉に行き着いたときのみ、直接点データにアクセスするので、探索に費やす時間が確実に削減される。

【0089】

(2) 最近傍点探索

(3) 近傍点リストの作成

(4) ニアレストネイバー法（巡回セールスマン問題）

【0090】

上記(2) (3) (4)の処理は、いずれも巡回セールスマン問題の演算高速化と密接に関係する処理である。特に、(3)、(4)の処理は、(2)を反復して用いるものであるので、まとめて説明する。

【0091】

(2) 最近傍点探索

この最近傍点探索は、図20に示す如く、領域内の注目点に最も近い点を探索する処理である。この処理は、(3)近傍点リストの作成、及び(4)ニアレストネイバー法において反復して用いられる。

【0092】

(3) 近傍点リストの作成

図20に示した如く、領域内の注目点について、その点に近い順に、見つかっ

た点を適当な数（リスト長と称する：図18の例では3個）分、列挙する。この処理は、最近傍点探索処理と、見つかった最近傍点をツリーから一時的に外す処理の反復により実現される。又、この処理は、巡回セールスマン問題の高速化のための各点データに付隨するデータ作成という位置付けの前処理であり、領域内全ての点に対して行う処理である。

## 【0093】

## (4) ニアレストネイバー法

このニアレストネイバー法は、図21に示す如く、初期点を適当に決めて注目点とし、「注目している点の未だ経路につなげられていない最近傍点を探し、発見された点を新たな注目点とする」という処理を、一巡経路となるまで繰り返す処理である。この処理は、最近傍点探索処理と、見つかった最近傍点をツリーから一時的に外す処理の反復により実現される。

## 【0094】

又、この処理は、高速な処理が行えることと、それなりの精度（巡回経路長）の解が得られるという2つの利点があるので、図22に示す巡回セールスマン問題の反復改善法の範疇に属する解法を用いるときの初期解（ステップ501）として用いることが出来る。

## 【0095】

前記最近傍点探索に際して、従来は、領域内の注目点について、領域内のその点以外の全ての点への距離を算出し、最も距離の近い点を挙げていた。

## 【0096】

又、前記近傍点リストの作成に際して、従来は、領域内の注目点について、領域内の該注目点以外の全ての点への距離を算出し、リストに距離の短いものから順にリスト長分に並べる処理をするようにしていた。

## 【0097】

又、前記ニアレストネイバー法を実現するための従来の方法は、各反復において、注目点について、未だ経路につなげられていない点全てについて距離を算出し、一番短いものを探し出し、注目点を更新するというものである。

## 【0098】

これらの処理は、全て(2)最近傍点探索を用いるものであるが、本発明による最近傍点探索の方法は、木のなぞり方により大きく分けて2種類ある。木の根から探索するトップダウン的方法、及び、最近傍点を探している注目点の存在する葉節点から探索するボトムアップ的方法である。

## 【0099】

まず、トップダウン的方法について、図23を参照して具体的な処理手順を説明する。

## 【0100】

この方法は、「現在の節点を根とする部分木のトップダウン的探索処理」を再帰的(recursive)に呼び出すことにより実行されるものである。即ち、破線で囲んだ部分が、「現在の節点を根とする部分木のトップダウン的探索処理」であるが、この処理の内部で、現在の節点を左もしくは右の節点に更新して破線部の処理を行う(ステップ1005、1006、1009、1010)。

## 【0101】

ステップ1001において、現在の節点を根に設定し、最近傍点迄の距離を充分大きな値に設定する。次いで「現在の節点を根とする部分木のトップダウン的探索処理」は、ステップ1002において、まず現在の節点が葉節点であるかどうかを判定する。葉節点である場合は、ステップ1003において、全ての点データにアクセスし、最近傍点が発見できれば更新する。内部節点ではある場合には、ステップ1004において、左右どちらの子節点の領域を優先して探索するかを決定する。この処理は、現在の節点の分割線の分割値(図18における、cutval)と、最近傍点を探している注目点の分割線の分割軸の値との比較により決定する。ステップ1007、ステップ1008の判定は、直前の節点で探索を行った節点の兄弟節点の表す領域を探索する必要があるかどうかの判定である。ここでは、図24に示す如く、最近傍点を探している注目点を中心とする、現在の最近傍点までの距離を半径に持つ円(ここで円とは、中心から等距離上にある点を結んだ閉曲線という意味であり、距離測度はユークリッド距離である必要はない)と、節点の表す領域が重なりを持つかどうか(含まれても良い)によって判定する。

## 【0102】

次いで、ボトムアップ的方法について、図25を参照して具体的な処理手順を説明する。

## 【0103】

この方法は、最近傍点を探している注目点の存在する葉節点を探索の開始節点に設定し、必要に応じて現在の節点の兄弟節点や親節点の探索を行う探索方法である（この場合、各接点に親節点を指すポインタを付加する必要が生じる。従って、例えば図18において、構造体の要素に、kdnode型ポインタ\*fatherを加える。ただし根節点の親節点は存在しないので、NULLとする）。

## 【0104】

ステップ2001において、現在の接点を最近傍点を探している注目点の存在する葉節点に設定し（従って、各点データは自分の存在する葉節点を指すポインタを持つ必要が生じる）、最近傍点までの距離を充分大きな値に設定する。ステップ2002は、実際には現在の接点は葉節点にあるので、現在の節点内の全ての点にアクセスし最近傍点を更新する。ステップ2003において、現在の節点の表す領域の外を探索する必要があるかどうかを判定する。具体的には、最近傍点を探している注目点を中心とする、現在の最近傍点までの距離を半径に持つ円が、現在の節点の表す領域の内部に含まれるかどうかを判定する。内部に含まれないなら、次いで破線により囲まれたループ処理に入る。

## 【0105】

ループからの脱出判定基準は2つある。一つはステップ2005における判定で、ここではステップ2006で更新された現在の節点が根であるかどうかを判定する。もう一つはステップ2008における判定で、ここでは現在の節点の親節点の表す領域の外を探索する必要があるかどうかを判定する。具体的には、ステップ2003と同じく、最近傍点を探している注目点を中心とする、現在の最近傍点までの距離を半径に持つ円が、親節点の表す領域に含まれているかどうかの判定である。

## 【0106】

ループ処理の内部のステップ2006において、現在の節点の兄弟節点の表す

領域を探索する必要があるかどうかを判定する。具体的には、最近傍点を探している注目点を中心とする、現在の最近傍点までの距離を半径に持つ円が、現在の節点の表す領域と重なりを持つかどうかの判定である。重なりを持つ場合は、ステップ2007において、実際に探索を行う。

#### 【0107】

即ち、図26に例示する如く、注目点が領域Dにあるとすると、ボトムアップ的方法では、探索を注目点Q1のある領域Dからスタートさせる（ステップ2001）。この領域において、全ての点について注目点Q1までの距離を調べ（ステップ2002）、発見された最近傍点がQ2であったとすると、次にこの領域外にQ2より近い点がある可能性があるかどうかを調べるため、Q1を中心とする、半径がQ1とQ2の間の距離の円が、領域Dの外にはみ出しているかを判定する（ステップ2003）。この例では、円がD内に収まっているので、まだ探索の必要がある。領域Dは根の表す領域（全体領域）ではないので（ステップ2005）、次に領域Dの兄弟である領域Eの探索が必要かどうかを判定する（ステップ2006）。判定の基準は円との重複であるが、この例では重複しているため、領域E内の探索を行う（ステップ2007）。この例では、領域EにおいてQ2よりもQ1に近い点、Q3が発見された。ここまでで領域D、E内（即ち、領域B内）の探索が終了したが、次に、領域Bの外にもQ3より近い点があるかもしれない、再び注目点Q1と最近傍点Q3との間の距離を半径に、注目点Q1を中心を持つ円が領域Bの外にはみ出しているかどうかを判定する（ステップ2008）。この例では、はみ出しており、領域Bの兄弟領域Cを探索する必要がある事が分かるので、現在の節点を、領域Bを表している節点に更新し（即ち、領域Dの親領域に更新し）（ステップ2009）、ステップ2005に戻る。以上のようなステップを繰り返して、この例では最終的には領域H内に最近傍点Q4を発見し、終了する。

#### 【0108】

##### （3）近傍点リストの作成

近傍点リストの作成は、上記（2）最近傍点探索と、「発見された最近傍点を木から一時的に外す」処理を交互に行うことにより実現される。

## 【0109】

木から一時的に点データを外すのは、次のようにして行うことができる。即ち、木の葉節点は、図18に示したように、各点データに付隨するインデックスの開始番号と終了番号を持っている。従って、基本的には、終了番号（又は開始番号）を1つ減らし（増やし）、減らされた番号（増やされた番号）の位置のデータと交換すればよい。もしその点を消すと、その葉節点内の点データが全て消えてしまうようなときには、その葉節点を訪れる必要を無くすように、自分の節点が表わす領域から点が消えている状態であれば、それが分かるように節点の情報を持つ（各節点に対し、点があるかどうかが分かるように、例えば図18において、構造体の要素にempty（整数型）を加える）。

## 【0110】

## (4) ニアレストネイバー法

このニアレストネイバー法の解法は、(3)と同様、上記(2)の最近傍点探索と、発見された最近傍点を木から一時的に外す処理を交互に行うことにより実現される。(3)と異なるのは、(3)は注目点では常に固定しているのに対し、(4)は、発見された最近傍点を注目点に更新していくところにある。

## 【0111】

従来、最近傍点探索については、注目点に対し、他の点データ全てにアクセスして距離算出演算をして、最大のものを残す必要があった。又、近傍点リストの作成については、各点データに対し、他の点データ全てにアクセスして距離算出演算をする他、各演算の度に現在のリストの相応しい順位に点データを挿入する必要があった。又、この作業は、全ての点データに対し行う処理なので、全ての点データの組に対し、距離算出演算を行っていた。又、ニアレストネイバー法の解法については、各点に対し、未だ巡回路につなげられていない点を認識し、それらの点データ全てにアクセスして距離算出演算を行い、最大のものを残すという対応をする必要があった。

## 【0112】

これらの方法は工夫が見られず、特に、近傍点リストの作成については、kd-木を使わないと、計算時間が膨大となるため、kd-木を作成しな

い方法、つまり巡回セールスマン問題の高速化のための前処理をせずに、いきなり巡回セールスマン問題を解くという方法を探らざるを得なくなっていた。しかしながら、前処理無しでは、限られた計算時間で精度のよい解を得ることはできず、加工時間短縮の効果は薄かった。

#### 【0113】

一方、本発明による  $k d$ -木による探索方法では、最近傍点探索については、トップダウン的な方法とボトムアップ的な方法の2種類があるが、特に、ボトムアップ的な方法は、注目点の存在する葉節点に直接アクセスできるので、大幅に高速化できる。

#### 【0114】

又、近傍点リストの作成については、特にツリーから点データを外す処理が高速でできるので、処理を大幅に高速化できる。この近傍点リストの作成は、巡回セールスマン問題の局所探索法の範疇に属する方法の高速化のための前処理であるので、間接的ではあるが、巡回セールスマン問題も高速化できる。

#### 【0115】

ニアレストネイバー法の解法についても、同様の理由で高速化できる。このニアレストネイバー法は、巡回セールスマン問題の局所探索法の範疇に属する方法の初期解であるので、結局、巡回セールスマン問題の解法処理の一部を高速化できることになる。

#### 【0116】

##### (5) 全一致問い合わせ

全一致問い合わせとは、ある点について、それと相対座標値が完全に一致するものが、領域の中に存在するかどうかについて問い合わせを行う処理である。この問い合わせを利用すれば、或る走査エリアと別の走査エリアにおける相対的な配置パターンが、全く一致しているかどうかを判定することができる。

#### 【0117】

全ての穴開け位置は、それぞれ所属する1つの走査エリアを持っている。走査エリア内の点は、巡回セールスマン問題を用いて訪問順序（ガルバノバス）が最適化される。巡回セールスマン問題の計算量は、他に比べて大きいので、全ての

走査エリアについて巡回セールスマントロUBLEを解く代わりに、全ての走査エリアについて、全ての点位置が全く等しいエリアを探す処理を行い、巡回セールスマントロUBLEを解く必要のないエリアを探す。そのようなエリアが発見でき、巡回セールスマントロUBLEを解く必要がなくなれば、計算高速化に直結する。

## 【0118】

巡回セールスマントロUBLEを解く必要を無くすという意味であれば、エリアに対する全ての点位置が全く等しいという必要ではなく、各点の相対的な位置関係が一致していれば充分である。即ち、図27上段のように、全ての点位置が全く等しい場合には、エリアPのガルバノバスを決定すれば、エリアQのバスを再計画する必要はない。同様に、図27下段に示すエリアRとエリアSのように、一方の点位置が他方の点配置にオフセットが載った状態である場合にも、やはり、エリアRのバスを決定すればエリアSのバスを計画する必要はない。オフセット量は、両エリア内の代表的な点、例えばx最小かつy最小な点同士の絶対座標の差にすればよい。

## 【0119】

この全一致問い合わせを利用した走査エリア同士の相対的な点配置一致判定処理は、エリア位置決定時に行われる。

## 【0120】

従来は、領域内の点全てにアクセスし、注目点の座標値と一致するかどうかを一つ一つ比較していた。

## 【0121】

これに対して、kd-木を用いる本発明では、ある注目エリアと別の注目エリアのオフセット量を計算し、ある注目エリアの各点の座標にオフセット量を加えた座標が別の注目エリアに存在するかどうかを、別の注目エリアのツリーの根節点から、現在の節点の子節点について、探している座標の存在する方の子節点へと次々に移るという方法により探索を行い、葉節点に到達したら、節点内の全ての点について、一致するかどうかを一つ一つ比較する。

## 【0122】

このようにして、全ての点データにアクセスするのではなく、二分木の根節点

からのトップダウン的探索方法により、比較の回数が減り、計算が高速になる。

【0123】

本発明で用いる  $k d$  - 木は、点位置データ、及び、点位置データが存在する領域から作成される。本実施形態においては、テンポラリな領域作成時とエリア位置確定時に  $k d$  - 木を作成する。

【0124】

即ち、テンポラリな領域作成時の直交領域探索処理は、「領域内の点を最少数の同寸矩形で囲む」処理において、エリア位置決定時のエリア内の点列挙その他に適用される。従って、最適L軸値決定処理によるループ呼び出しで、一時的な領域を作成する度に、その一時的な領域及び一時的な点位置データに関する木を形成する。

【0125】

又、前記最近傍点探索処理、近傍点リストの作成、及び、ニアレストネイバ法の処理は、いずれも、巡回セールスマン問題に関するものである。巡回セールスマン問題は、大きく分けて、エリア位置訪問順序（ステージパス）と、走査エリア内穴開け位置訪問順序（ガルバノパス）を最適化するものである。故に、エリア位置確定時に、最適L軸値での一時的な領域及びL軸値という点データに関する木を1つ作成すると共に、走査エリア領域と走査エリアに従属する穴開け位置という点データに関する木を、各走査エリアについてそれぞれ1つずつ作成する。この各走査エリアに関する木は、前記全一致問い合わせにも利用される。

【0126】

なお、前記説明においては、エリアの形状が矩形とされていたが、エリアの形状はこれに限定されず、円形でもよい。

【0127】

又、走査エリアに対してレーザビームを走査する手段もガルバノスキャナによるものに限定されず、例えば出願人が特開2000-71089や特開平11-144358、特開平10-245275で提案したような、リニアモータXYステージと高速加工ヘッドを組み合わせたハイブリット加工システム（いわゆるスクリーンカットシステム或いはフラッシュカットシステム）であってもよい。

## 【0128】

又、適用対象も、レーザ穴開け機に限定されず、レーザビーム以外の加工手段を用いた一般の加工機（例えば機械式ドリルによる穴開け装置）にも同様に適用できることは明らかである。更に、ハードウェアとは分離して、例えばインターネットを利用し、データを受取って計算結果を返す計算サービスに用いることも可能である。

## 【0129】

## 【発明の効果】

本発明によれば、加工時間短縮を目的とする加工計画の精度を落とすことなく、計画にかかる計算時間を短縮することができる。

## 【0130】

本発明より高速化される処理及び使用回数は、例えば前記直交領域探索においては、一時的領域内の一つ一つのエリア決定の途中、エリア決定時に用いられるので、凡そ「一時的領域の作成回数×エリア数平均」回である。

## 【0131】

又、前記近傍点リストの作成は、巡回セールスマン問題の前処理であるため、全ての点データ及びエリア位置データに各1回ずつ用いられるので、「穴開け位置数+半領域のエリア数」回である。

## 【0132】

又、前記ニアレストネイバー法は、巡回セールスマン問題につき1回ずつ用いられるので、最大「1+基板全体のエリア数」回である（“1”はステージパスを計画する時の巡回セールスマン問題）。

## 【0133】

又、前記全一致問い合わせにおけるエリア同士の比較の回数は、最大約「(基板全体のエリア数)<sup>2</sup>/2」回である（二乗の項が存在するので計算量が多いよう一見感じられるが、エリア同士の穴配置の完全一致判定は、一点でも全一致点が存在しない点が存在すれば判定終了なので、一般的には大した計算量とはならない）。

## 【0134】

上記のうち、特に、近傍点リストを作成し巡回セールスマントークン問題を解くという流れの処理（反復改善法の範疇に属する解法は基本的にこの流れに沿う）、全ての計算に対して占める割合が多く、その分、高速化が顕著に表われる。

## 【0135】

一方、木の作成にかかる時間は余計に必要となるが、木の作成回数は、「一時的領域の作成回数 + 1 + 基板全体のエリア数」回であり、木の作成により高速化する処理を行う回数よりも十分小さく、又、作成に費やす時間もそれほど長くはないので、問題にならない。

## 【0136】

従って、全体的には計算時間の短縮効果は十分にある。

## 【図面の簡単な説明】

## 【図1】

本発明が適用されるレーザ穴明け機の全体構成を示す斜視図

## 【図2】

同じく詳細構成を示す斜視図

## 【図3】

同じくステップアンドリピートによる全体の加工動作を示すタイムチャート

## 【図4】

同じく各走査エリア内の加工動作を示すタイムチャート

## 【図5】

レーザ穴開け機の基本的なシステム構成の例を示すブロック図

## 【図6】

レーザ穴開け機の加工計画装置の構成及び処理手順を示す図

## 【図7】

前記加工計画装置によるユニット間距離設定の原理を説明するための平面図

## 【図8】

同じくビーム走査回数及びステージ移動回数の重み付けした和と、最適なユニット間距離の関係の例を示す線図

## 【図9】

同じくエリア重複時の穴開け位置の配分の原理を説明するための平面図

【図10】

同じく穴開け位置配分の例を示す線図

【図11】

同じく穴開け位置の広がりの中心でエリア位置を微調整している様子を示す平面図

【図12】

同じく、始点、終点の最適設定に関して、一巡経路を決定後、最長の移動を除いている様子を示す平面図

【図13】

同じく「（一巡経路）－（最長の移動）」により一巡経路を決定している様子を示す平面図

【図14】

同じく左右ユニットの移動パターンのマッチングの原理を示すタイムチャート

【図15】

ツリーの節点同士の関係を表す用語を説明するための線図

【図16】

本発明で用いられる2次元点データの例を示す平面図

【図17】

同じくツリーへの格納状態を示す線図

【図18】

同じくデータの記述例を示す線図

【図19】

本発明の適用対象である矩形領域探索の様子を示す平面図

【図20】

同じく最近傍点探索及び近傍点リストの作成の様子を示す平面図

【図21】

本発明の実施形態における、ニアレストネイバー法による巡回経路の作成過程の例を示す平面図

【図22】

同じく反復改善法（局所探索法）による処理手順の例を示す流れ図

【図23】

同じくトップダウン的方法による処理手順の例を示す流れ図

【図24】

前記トップダウン的方法における、探索を行うべき領域を判定する様子を示す平面図

【図25】

本発明の実施形態における、ボトムアップ的方法による処理手順の例を示す流れ図

【図26】

前記ボトムアップ的方法における、探索を行うべき領域を判定する様子を示す平面図

【図27】

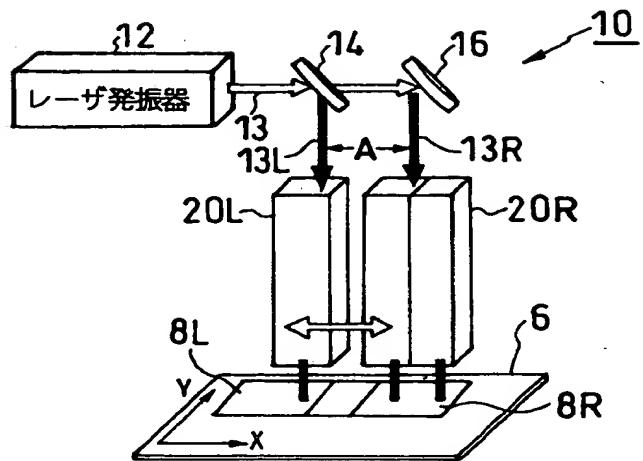
本発明の実施形態における、全一致問い合わせの有効性を示す平面図

【符号の説明】

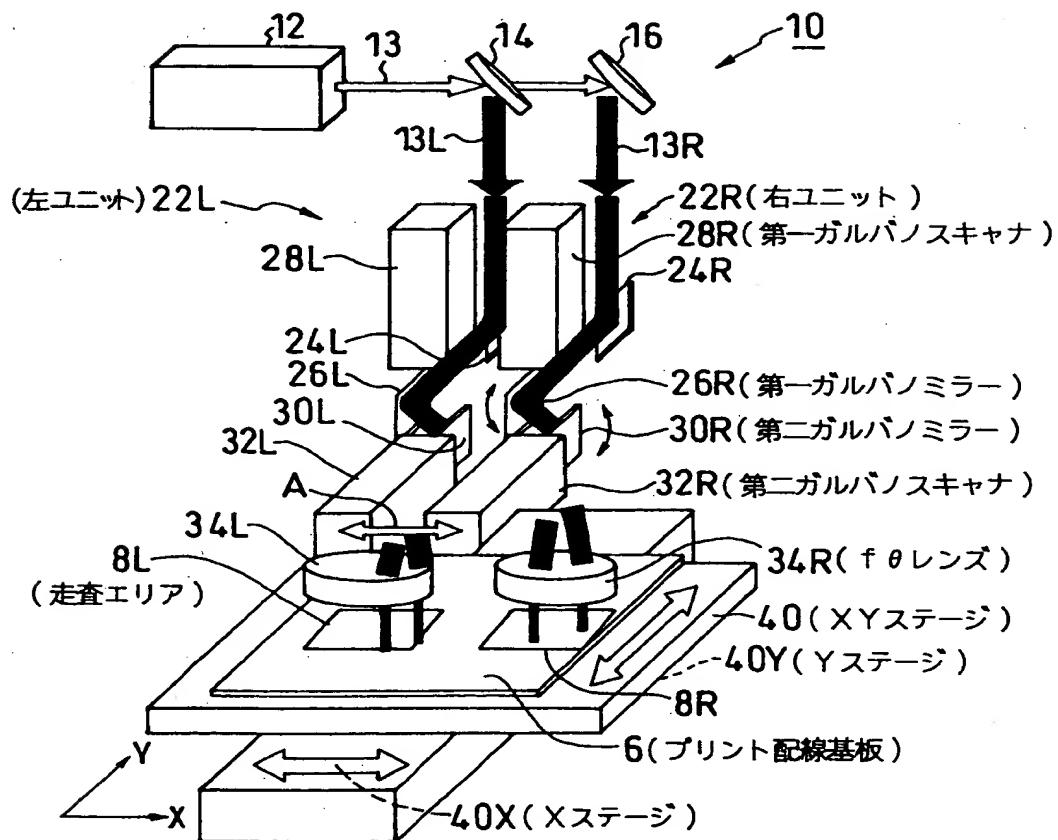
- 6 … プリント配線基板（ワーク）
- 8、 8'、 8 L、 8 R … 走査エリア
- 1 0 … レーザ穴開け機
- 1 2 … レーザ発振器
- 1 3、 1 3 L、 1 3 R … レーザビーム
- 2 2 L、 2 2 R … ガルバノユニット
- 4 0 … XYステージ
- 5 0 … 入力データ
- 5 2 … 加工計画装置
- 5 4 … 変換データファイル群
- 6 0 … 穴開け加工実処理部
- 6 2 … 制御装置
- 6 4 … 内部機器

【書類名】 図面

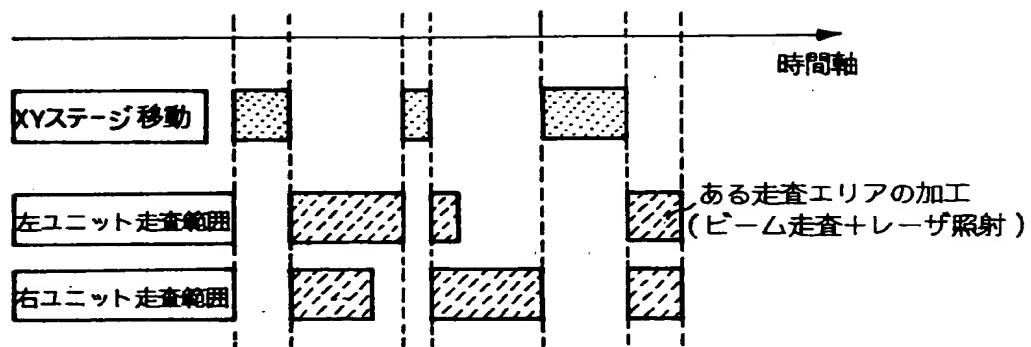
【図1】



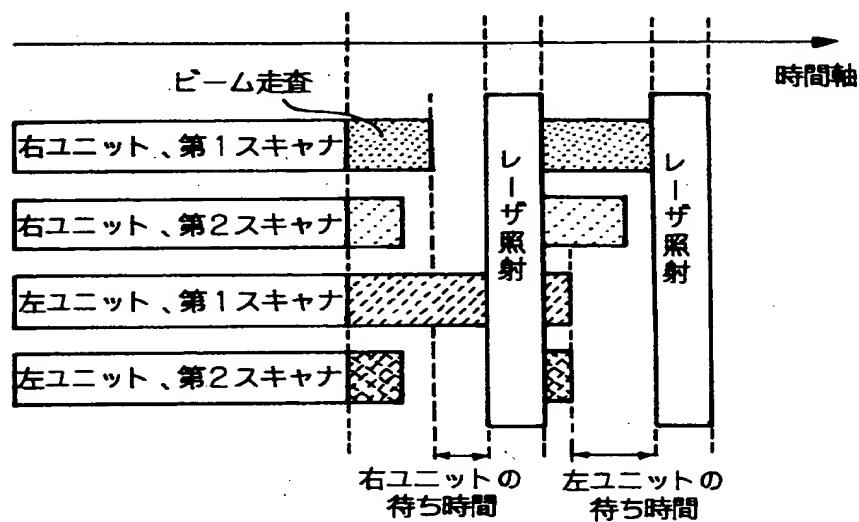
【図2】



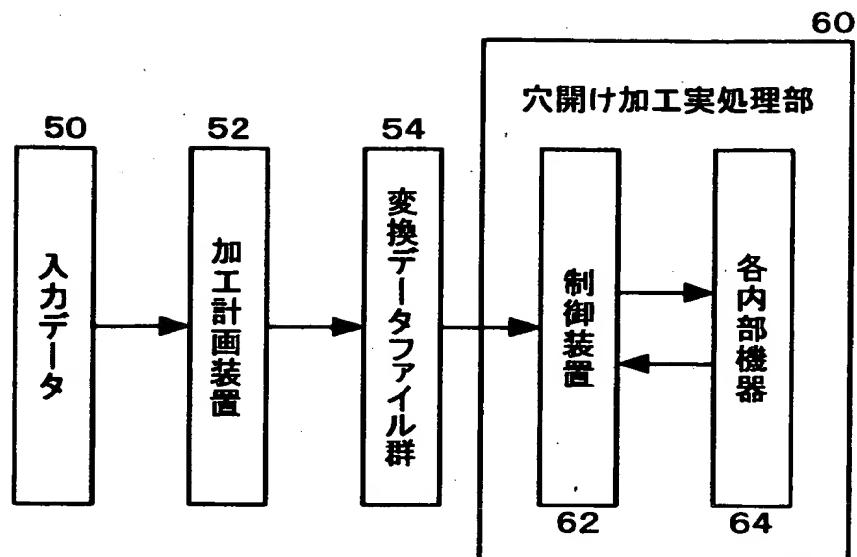
【図3】



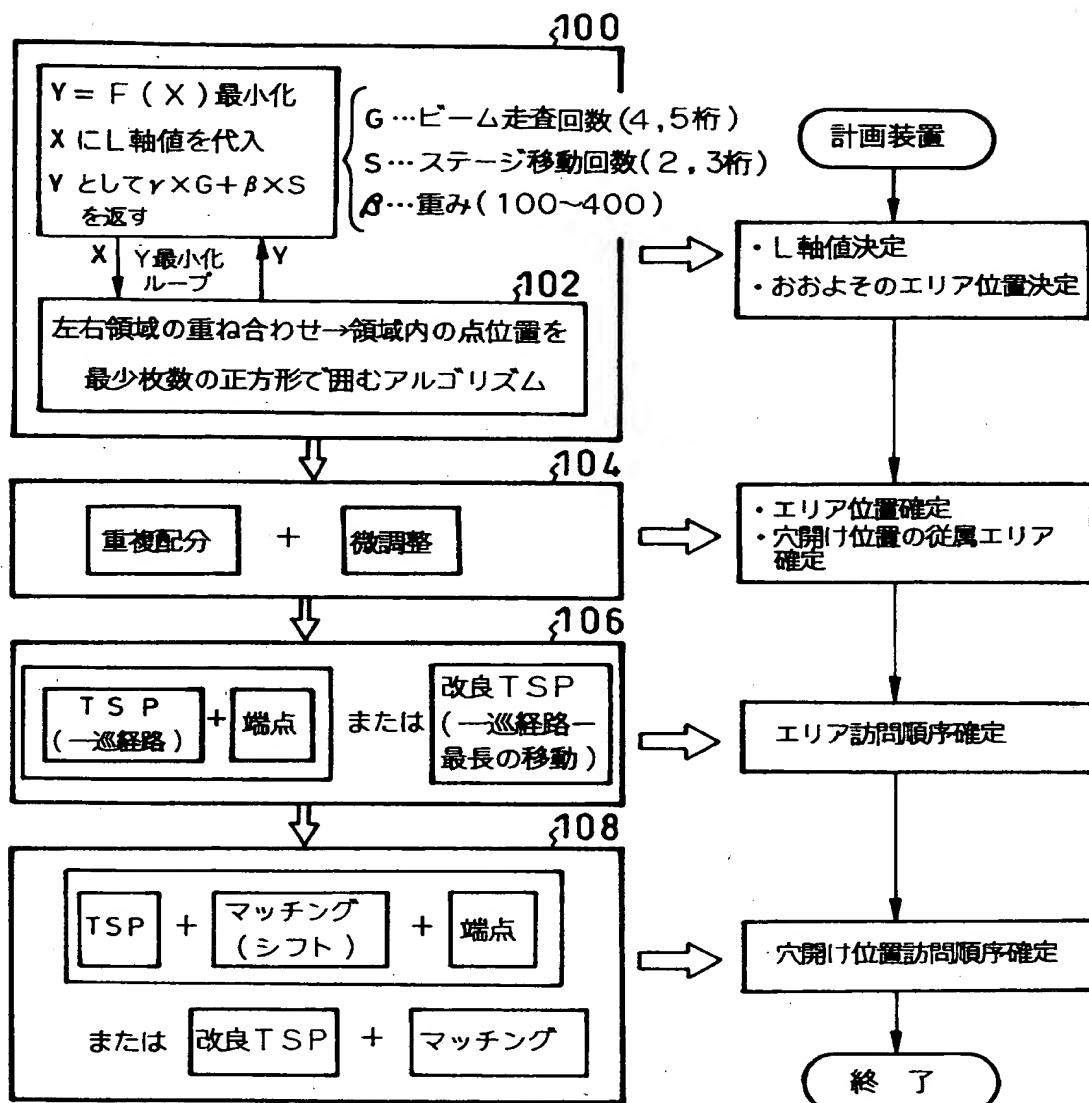
【図4】



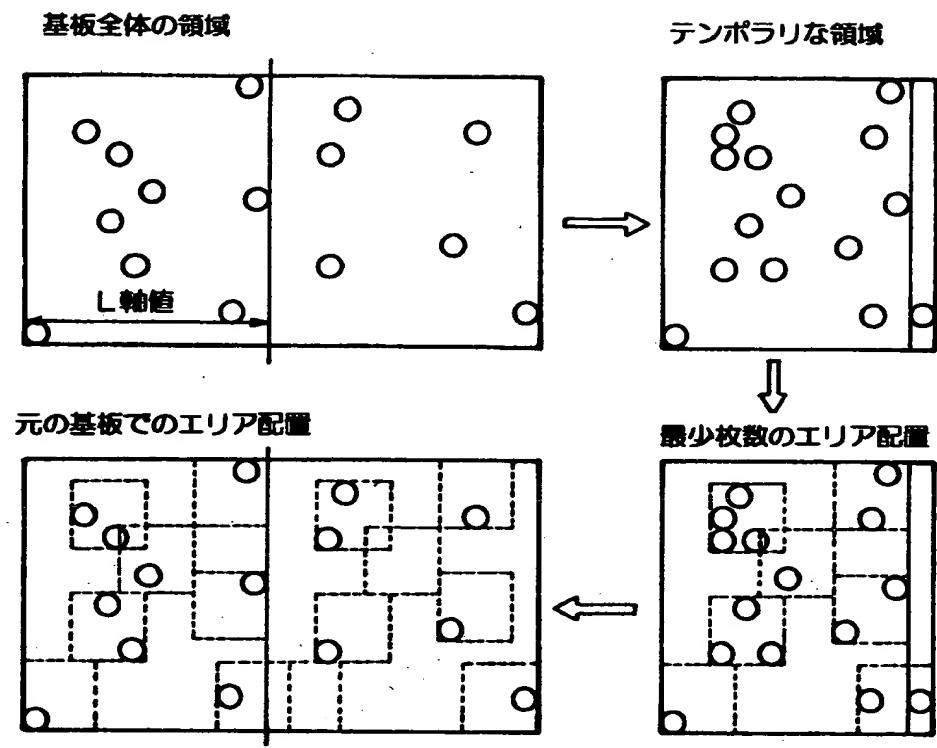
【図5】



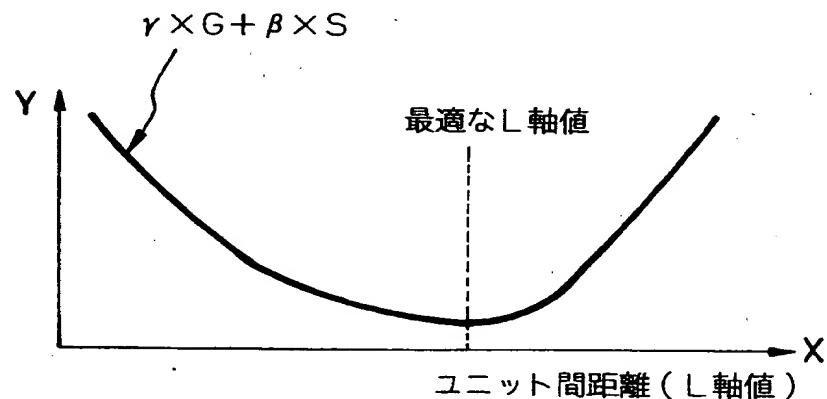
【図6】



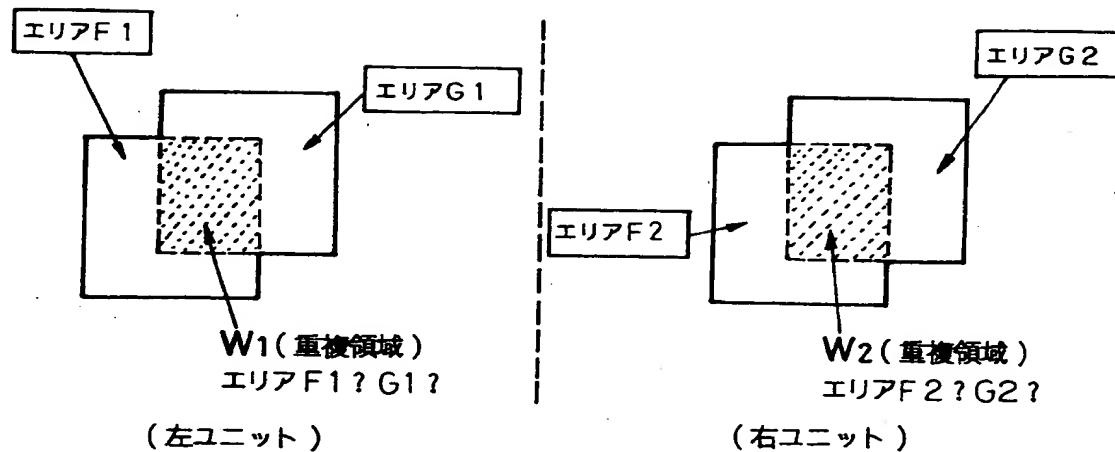
【図7】



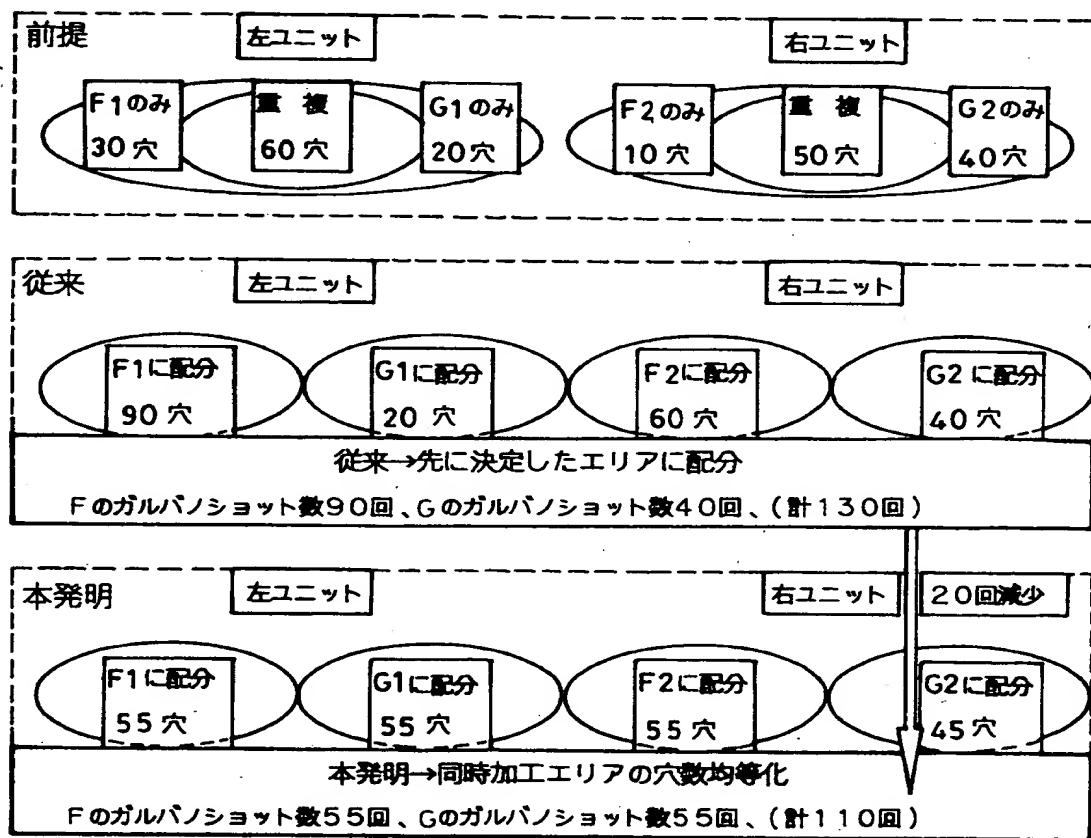
【図8】



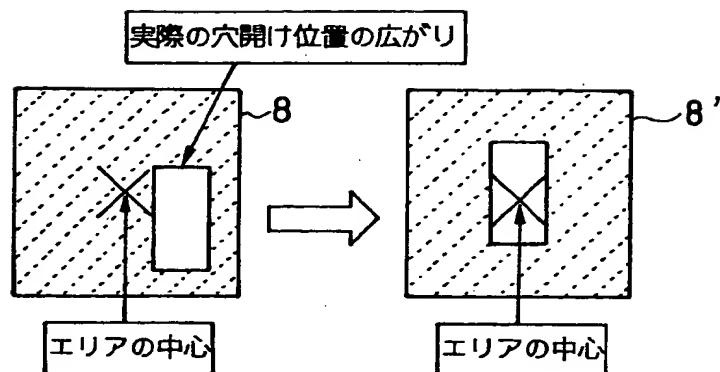
【図9】



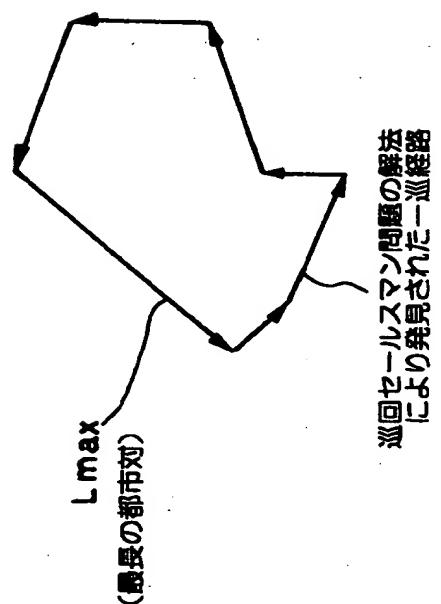
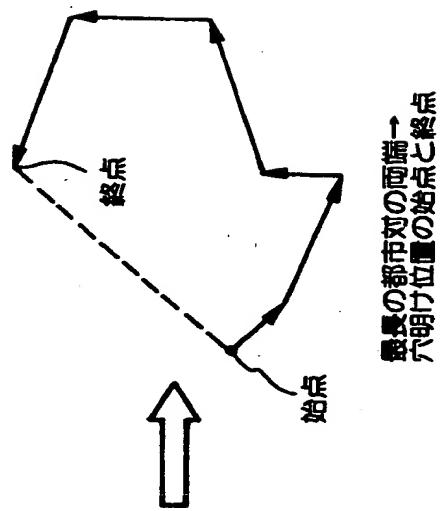
【図10】



【図11】



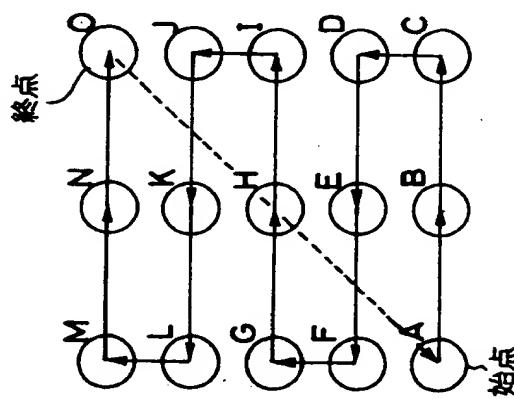
【図12】



【図13】

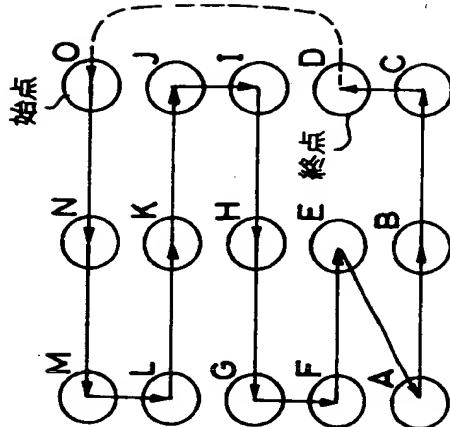
格子状の都市配列の場合（格子間隔1）

1つの最適バス（距離14）
（一巡回）は最長



バス長は増加（約14.4）
---------------

“巡回経路長”を“（一巡回）”にしておけば交換無し



2オプト法  
(TSPの典型例)

2本の枝を付け替えて  
巡回経路長が短くなる  
なら更新（AO, DEと  
AE, DOを交換）

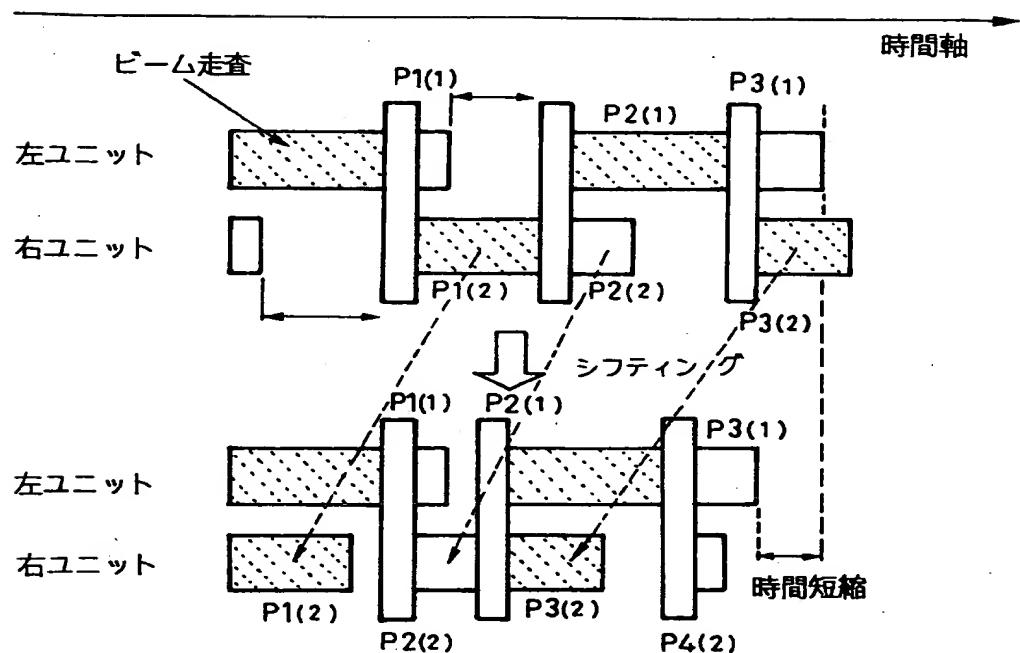
巡回経路長は、約17.4に減少



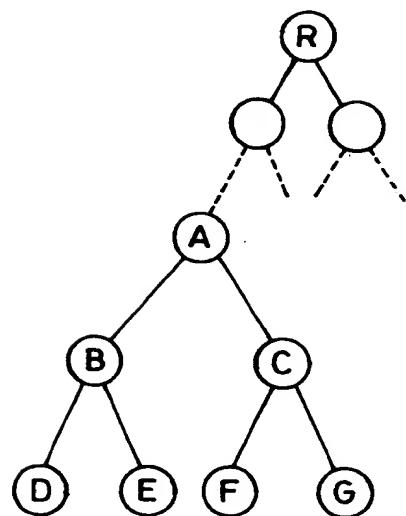
巡回経路長は、約18.5



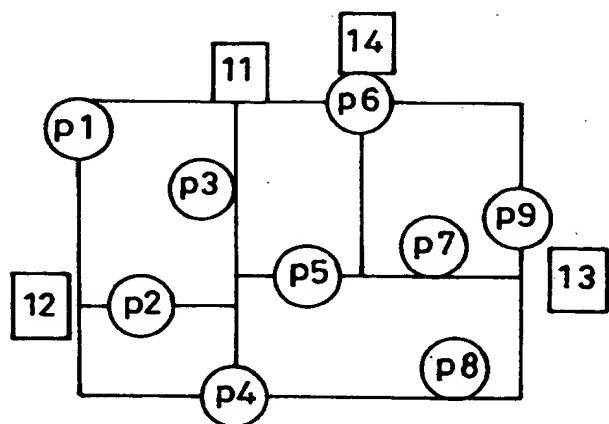
【図14】



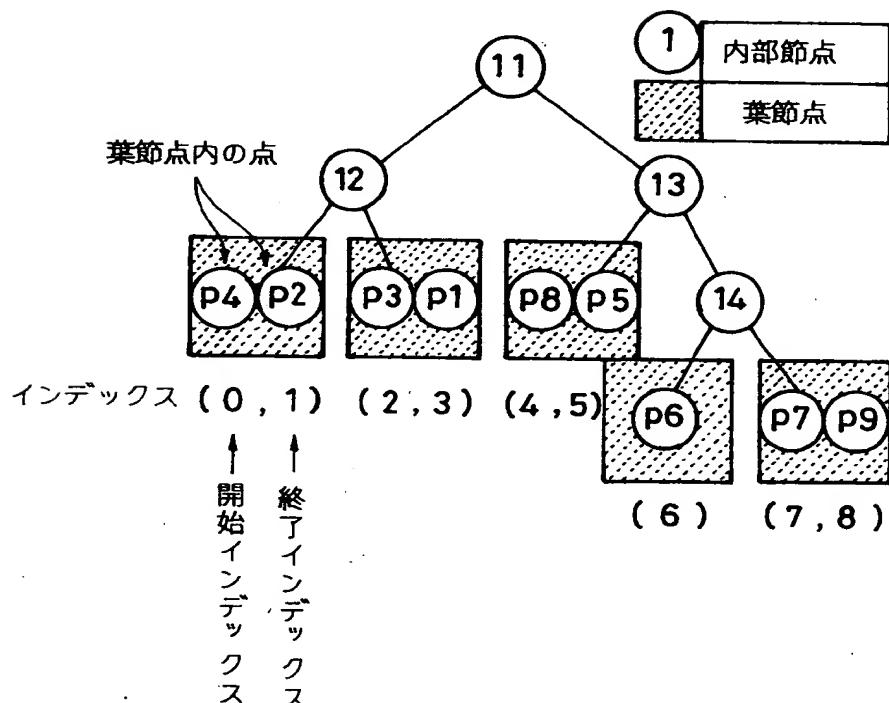
【図15】



【図16】



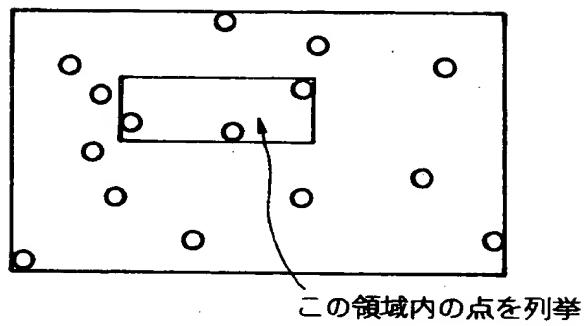
【図17】



【図18】

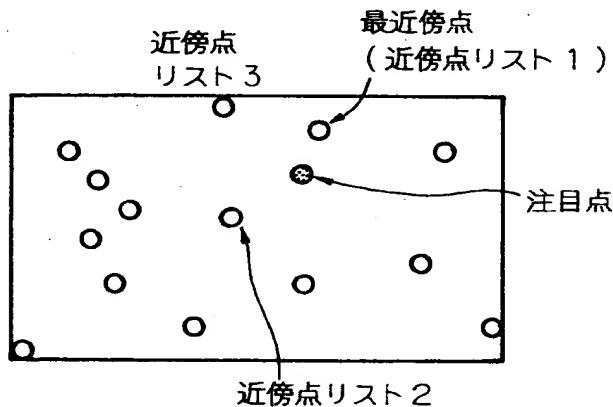
```
struct kdnnode {
    int      bucket;
    int      cutdim;
    float    cutval;
    kdnnode *loson, *hison;
    int      lopt, hipt;
};
```

【図19】

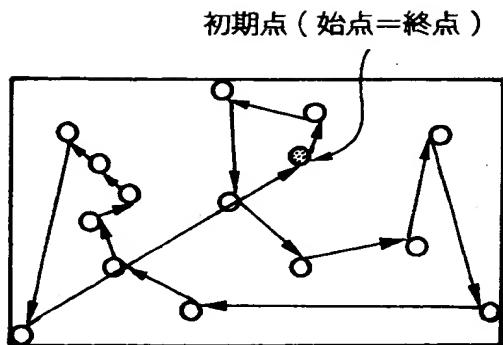


この領域内の点を列挙

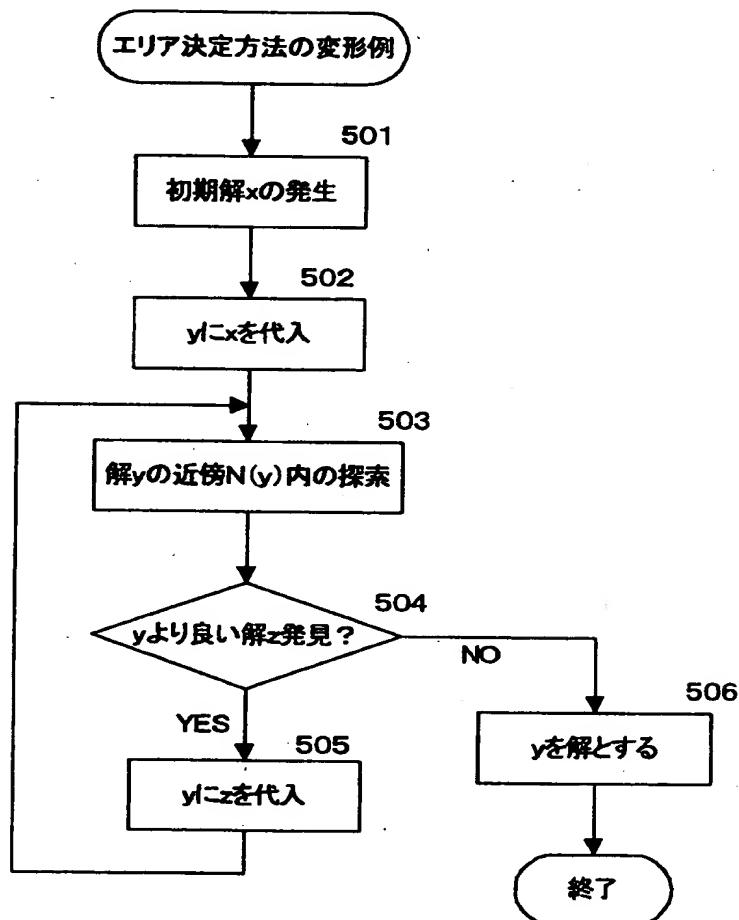
【図20】



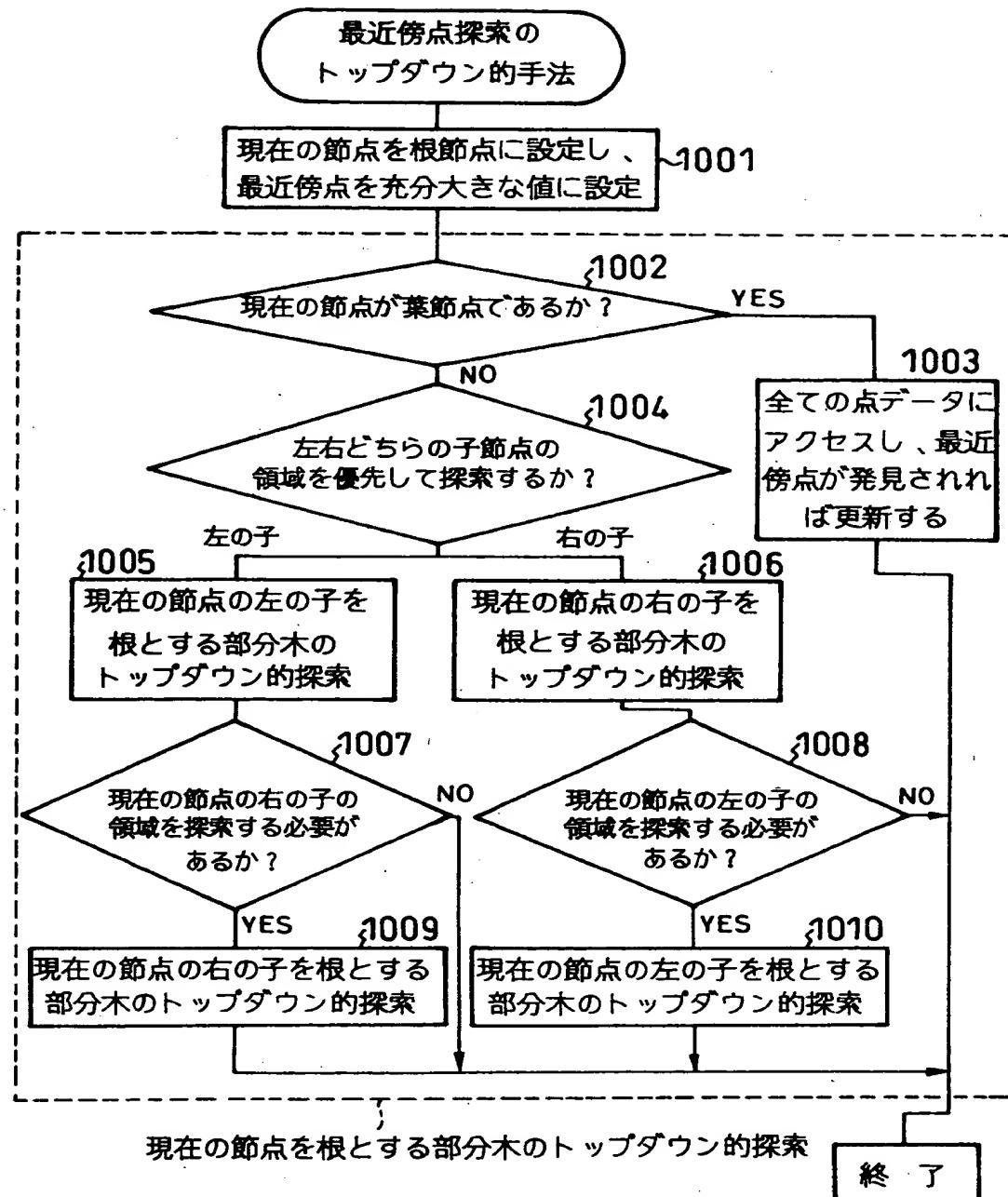
【図21】



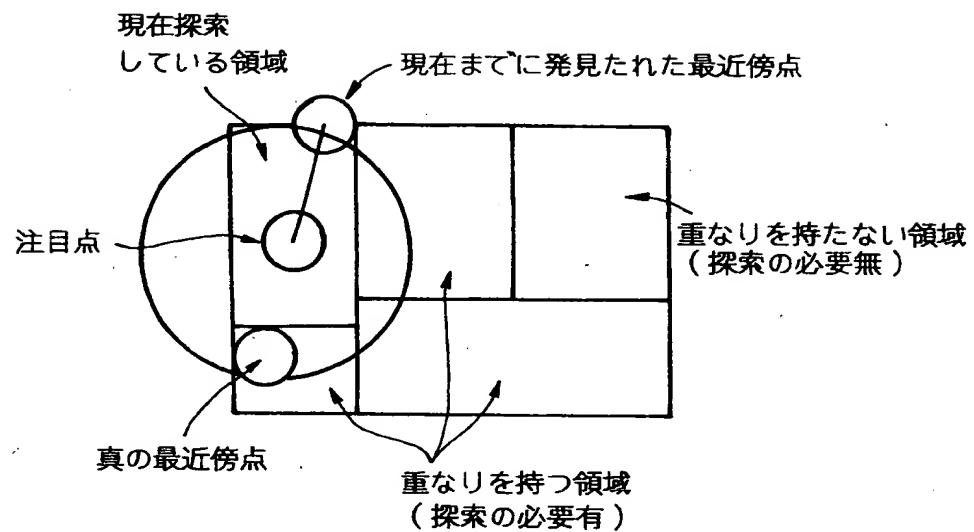
【図22】



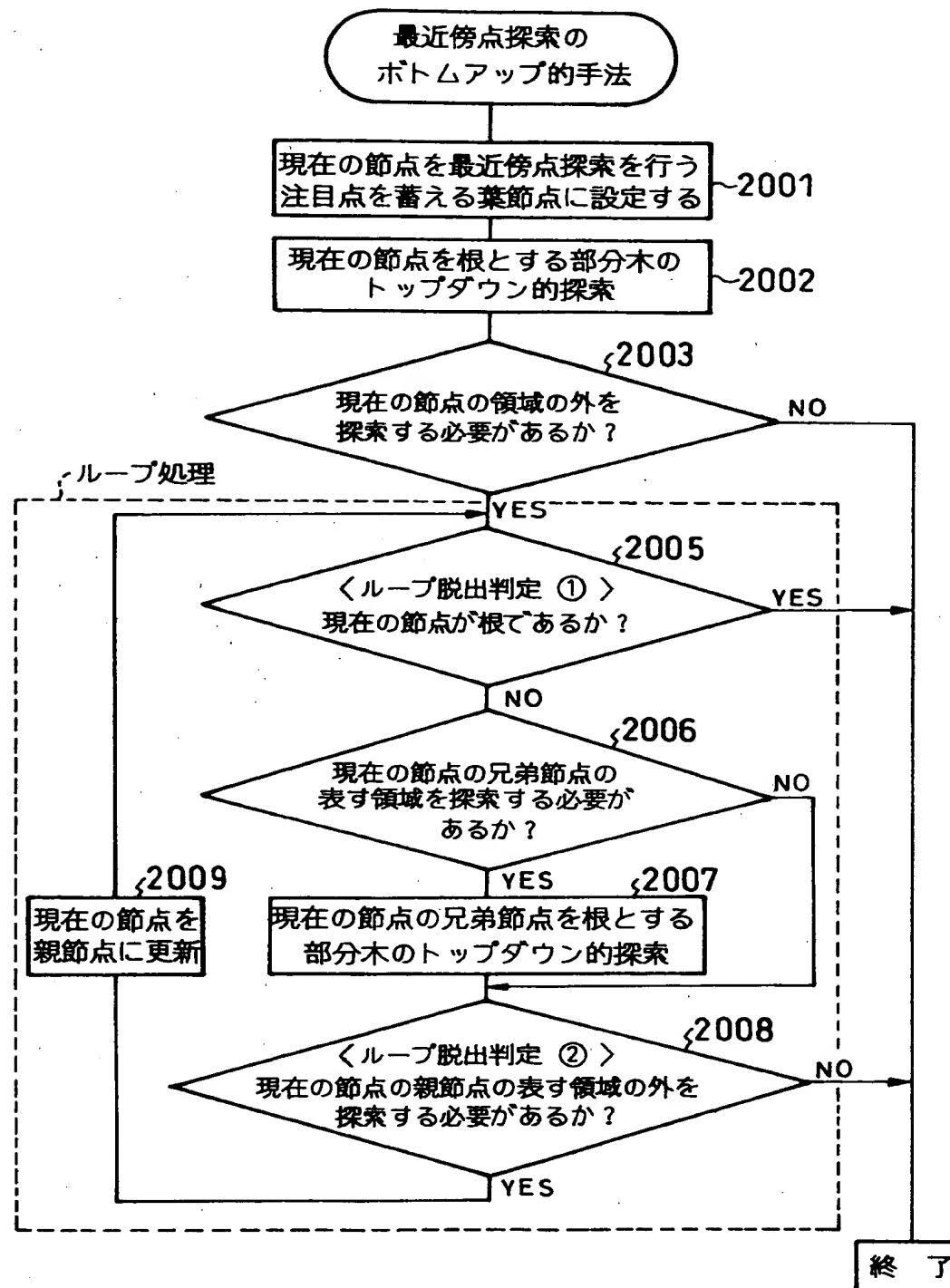
【図23】



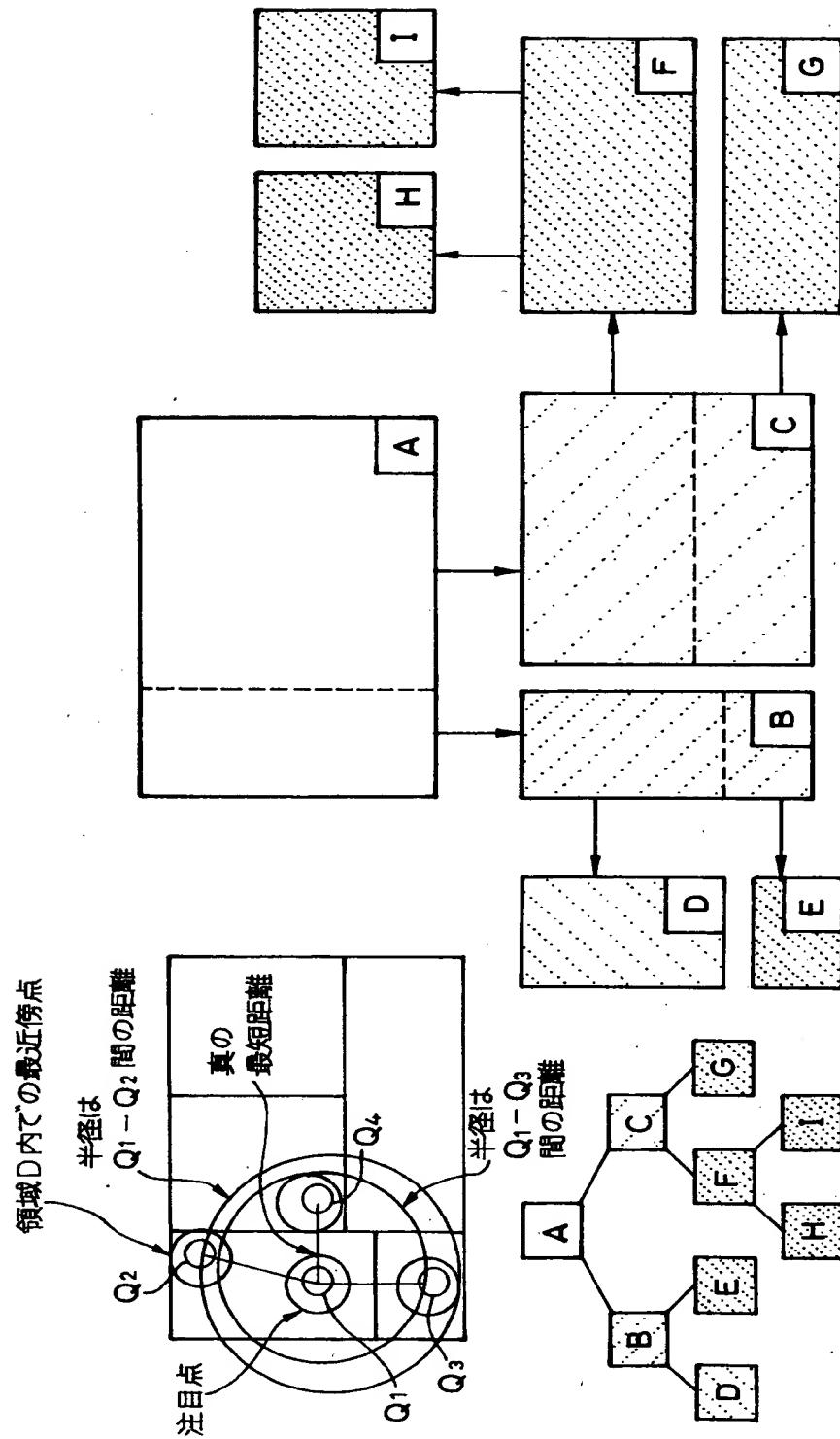
【図24】



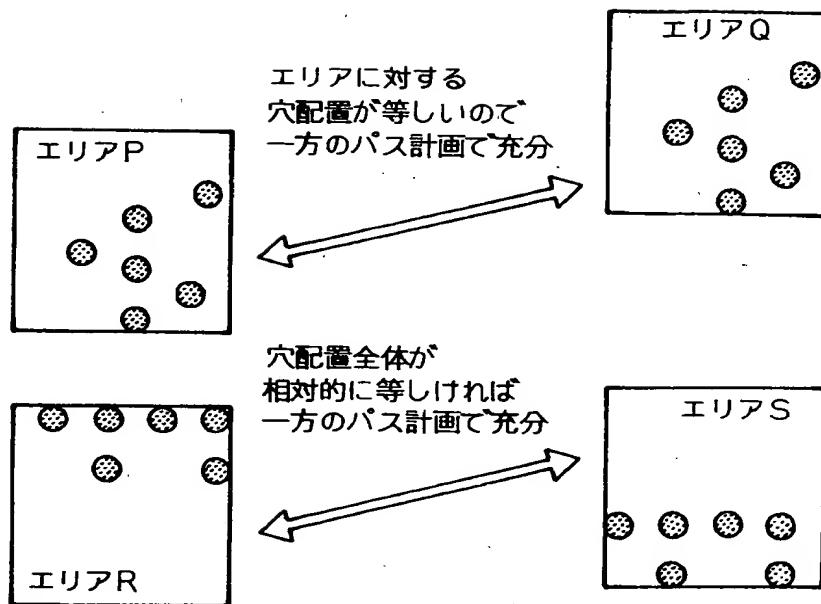
【図25】



【図26】



【図27】



【書類名】 要約書

【要約】

【課題】 加工時間短縮を目的とする加工計画の精度を落とすことなく、計画にかかる計算時間を短縮する。

【解決手段】 ワーク上に散在する加工位置又は加工エリアを表わす点の位置データを、 $k$  次元二分探索木によるツリー型のデータ構造で表現する。

【選択図】 図17

出願人履歴情報

識別番号 [000002107]

1. 変更年月日 1994年 8月10日

[変更理由] 住所変更

住 所 東京都品川区北品川五丁目9番11号

氏 名 住友重機械工業株式会社